Optimally refined isogeometric analysis

Daniel Garcia[1], Michael Bartoň[1], and David Pardo[2,1,3]

[1] BCAM– Basque Center for Applied Mathematics,
Alameda de Mazarredo 14, 48009 Bilbao, Basque Country, Spain
dgarcia@bcamath.org, mbarton@bcamath.org
[2] Department of Applied Mathematics, Statistics, and Operational Research
University of the Basque Country UPV/EHU, Leioa, Spain
[3] Ikerbasque (Basque Foundation for Sciences), Bilbao, Spain
dzubiaur@gmail.com

Abstract
Performance of direct solvers strongly depends upon the employed discretization method. In particular, it is possible to improve the performance of solving Isogeometric Analysis (IGA) discretizations by introducing multiple C^0-continuity hyperplanes that act as separators during LU factorization [8]. In here, we further explore this venue by introducing separators of arbitrary continuity. Moreover, we develop an efficient method to obtain optimal discretizations in the sense that they minimize the time employed by the direct solver of linear equations. The search space consists of all possible discretizations obtained by enriching a given IGA mesh. Thus, the best approximation error is always reduced with respect to its IGA counterpart, while the solution time is decreased by up to a factor of 60.

Keywords: solver-based discretization, continuity-aware optimal dissection, direct solvers, multi-frontal solvers, refined IsoGeometric Analysis (rIGA)

1 Introduction

When modeling physical and/or engineering processes, computational issues are crucial [9, 5, 3, 11, 7]. Consider, for example, simulations of turbulent flows around wind turbines [13] or simulations of tectonic activities on a large scale of tens of kilometres with required accuracy within several centimetres. To simulate these complex processes with sufficient accuracy, typically by using Finite Elements (FE) or IsoGeometric Analysis (IGA), requires a large number of unknowns.

Direct solvers are a popular option to solve linear systems arising from FE or IGA discretizations. While FE systems employ a large number of unknowns compared to IGA, the latter method exhibits a suboptimal behavior in terms of computational time per solved unknown (see [4]).

It was recently shown in [8] that an IGA discretization enriched with properly designed C^0 continuity hyperplanes (that act as separators in terms of the direct solver) offers a superior alternative than both FEM and IGA in terms of computational time required by the direct solver for a given fixed mesh topology. This strategy was denoted as *refined Isogeometric Analysis (rIGA)*.

While rIGA is based on the enrichment of a highly continuous C^{p-1} discretization (with p being the polynomial degree) via C^0 hyperplanes, in here we further expand this search to the more general case of C^q hyperplanes (with $0 \leq q \leq p-1$). We denote the resulting discretization as *Optimally refined Isogeometric Analysis (OrIGA)*.

The rest of the paper is organized as follows. Section 3 introduces the concept of rIGA in 2D with the focus on separators of arbitrary continuities. Section 4 discusses connections between rIGA and OrIGA and proposes a fast algorithm to compute OrIGA discretizations. The numerical results are shown in Section 5. Finally, the paper is concluded in Section 6.

2 Model Problem and Notation

In this work, for simplicity we assume a discretization of a square 2D mesh $\mathcal{M}$ consisting of $m^2 = 2^s \times 2^s$ elements, s being the number of subdivision levels. We define a bivariate tensor product spline space over $\mathcal{M}$ and denote by p^x and p^y the degrees of the univariate splines in x and y directions, respectively. C^q hyperplanes preserve the tensor product structure (that is, we do not consider, e.g., T-junctions).

We aim to minimize an estimate of the LU factorization cost. We assume the mesh is repeatedly bisected using separators across the y and then the x direction, see Fig. 1. While conceptually it would be possible to seek also for the optimal positions of the separators, in this work we focus only on finding the optimal continuities. These separators dictate the order of LU factorization (by following an LU elimination ordering inverse to that of the introduced separators).

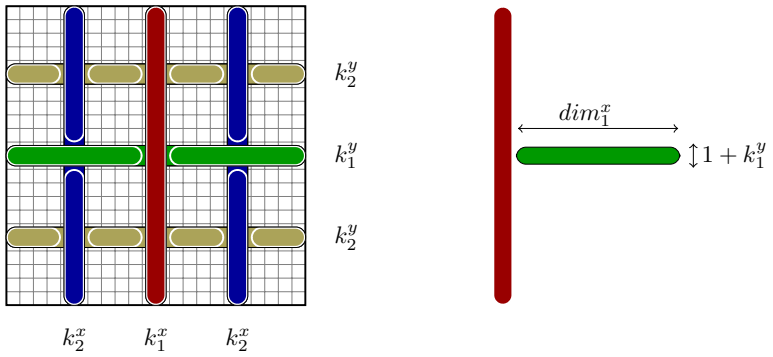


Figure 1: Left: two subdivision levels of a 2D mesh. We seek those continuities $(k_1^x, k_2^x, k_1^y, k_2^y)$ of the separators that minimize the total cost of LU factorization. Right: the number of basis functions of one horizontal separator in the first subdivision level (green) is $dim_1^x(1 + k_1^y)$. Decreasing the continuity k_1^y of the green separator increases the dimension of the orthogonal spline space (red) by $p^y - (1 + k_1^y)$.

For simplicity, we assume that the continuities of all separators at the same subdivision level are equal. That is, let $\{k_1^x, \ldots, k_s^x\}$ and $\{k_1^y, \ldots, k_s^y\}$ be the unknown continuities up to the s-th

subdivision level in x and y directions, respectively. We write

$$\mathbf{k} = (\mathbf{k}^x, \mathbf{k}^y) = (k_1^x, \ldots, k_s^x, k_1^y, \ldots, k_s^y), \tag{1}$$

see Fig. 1. In contrast to [8], in here we consider separators of any possible continuity.

3 Minimization Problem

OrIGA discretization is defined as the one that minimizes the number of floating point operations (FLOPs) used by the LU factorization for a fixed mesh topology and order of approximation p. More precisely, it is given as the solution argument of the following minimization problem

$$\arg\min_{\mathbf{k}\in S} F(\mathbf{k}), \tag{2}$$

where $F(\mathbf{k})$ is the cost functional, and S is the search space. In subsection 3.1 we define $F(\mathbf{k})$, while subsection 3.2 is devoted to describe S.

3.1 Minimization functional

Following [8, 4], we realize that the cost of the LU factorization is dominated by the cost of eliminating the top dense matrices at each level. Thus, neglecting the remaining contributions, the cost of the factorization grows in a cubic fashion with the number of degrees of freedom (multivariate spline basis functions) of each separator. Approximating the number of FLOPs of all the factorizations during the mesh dissection, the minimization functional reads as

$$F(\mathbf{k}) = \sum_{i=1}^{s} \#cuts_i^y (dim_i^y (1+k_i^x))^3 + \#cuts_i^x (dim_i^x (1+k_i^y))^3 \tag{3}$$

where $\#cuts_i^y$ is the number of cuts in the y-direction inserted in the i-th subdivision level, we sum over all the separators in both directions, and the summands are cubes of dimensions of the tensor product spline spaces associated to the separators. That is, dim_i^x and dim_i^y are the dimensions of the univariate spline spaces in the direction of the separator.

Note that for s subdivision levels, we have 2^s-1 separators in every direction. This generates 2^{2s} sub-meshes, and the number of boundary spline spaces that separate them grows also exponentially. Assuming the first separator is vertical (associated with an unknown continuity k_1^x), the number of boundary cuts in the i-th subdivision level is

$$\begin{aligned} \#cuts_i^y &= 4^{i-1}, \\ \#cuts_i^x &= 2\cdot 4^{i-1}. \end{aligned} \tag{4}$$

The dimensions of univariate spline spaces associated to the cuts in the i-th level are

$$\begin{aligned} dim_i^y &= \frac{m}{2^{i-1}} + p^y + \sum_{j=1}^{s-i+1} 2^{j-1}(p^y - (k_{i+j-1}^y + 1)), \\ dim_i^x &= \underbrace{\frac{m}{2^i} + p^x}_{(i)} + \underbrace{\sum_{j=1}^{s-i} 2^{j-1}(p^x - (k_{i+j}^x + 1))}_{(ii)}, \end{aligned} \tag{5}$$

where the first term (i) corresponds to the dimension of the spline space if there were no separators, and (ii) is the number of degrees of freedom that were added to the space by all the complementary (orthogonal) separators that intersect the space under consideration, see Fig.1 b).

For example for $s = 2$, $i = 1$, the dimension of the spline space associated to the first vertical cut (red in Fig. 1) becomes

$$dim_1^y = m + 4p^y - k_1^x - 2k_2^x - 3, \tag{6}$$

since all horizontal separators intersect it.

For general s, by substituting (4) and (5) into (3), we obtain

$$\begin{aligned} F(\mathbf{k}) &= \sum_{i=1}^{s} 4^{i-1} \left(\frac{m}{2^{i-1}} + p^y 2^{s-i+1} - \sum_{j=1}^{s-i+1} 2^{j-1}(k_{i+j-1}^y + 1) \right)^3 (1 + k_i^x)^3 \\ &+ 2 \cdot 4^{i-1} \left(\frac{m}{2^i} + p^x 2^{s-i} - \sum_{j=1}^{s-i} 2^{j-1}(k_{i+j}^x + 1) \right)^3 (1 + k_i^y)^3. \end{aligned} \tag{7}$$

We see that F contains (sixtic) terms of the form $(k_i^x \cdot k_j^y)^3$ with both positive and negative factors. Therefore, the minimizer of F is non-trivial and cannot be in general computed analytically. The objective function (7) is a generalization of (1) in [8], where now the continuities of the separators may be different from zero.

The cost functional F is completed by adding the cost that comes from static condensation [12], namely:

$$F_{SC} = 2^{2s} \left(\left(\frac{m}{2^s} + p^x - 1 \right) \left(\frac{m}{2^s} + p^y - 1 \right) \right)^3, \tag{8}$$

and it does not depend on $\mathbf{k}$. The complete functional cost is expressed as

$$F_T(\mathbf{k}) = F_{SC} + F(\mathbf{k}) \tag{9}$$

3.2 Search space and its reduction

Our search space S is defined as a discrete set of possible continuities of separators in all subdivision levels, that is,

$$S = \{\mathbf{k}, k_i^z = 0, \dots, p^z - 1,\ z = x, y,\ i = 1, \dots, s\}. \tag{10}$$

This space is a generalization of the space of admissible continuities in rIGA, see Fig. 2.

We seek a minimizer that lies in $\mathbb{N}^{2s}$. Thus, our search space is *finite*. Unfortunately, the number of combinations grows exponentially with the number of subdivision levels s. Namely, we have

$$\#combinations = (p^x p^y)^s. \tag{11}$$

Therefore an exhaustive search is not possible even for a moderate value of s. It is necessary to restrict the search space. First, we notice that $k_1^x = 0$ since it appears in (7) only once and with positive sign. Additionally, the sequence of optimal continuities in both directions has to be non-decreasing, which is formalized in the following lemma.

Lemma 1. *The vector of continuities* $\mathbf{k}$ *of Eq.* (1) *that minimizes Eq.* (7) *satisfies*

$$k_i^z \leq k_{i+1}^z \quad \textit{for all} \quad z = x, y \quad \textit{and} \quad i = 1, \dots, s-1. \tag{12}$$

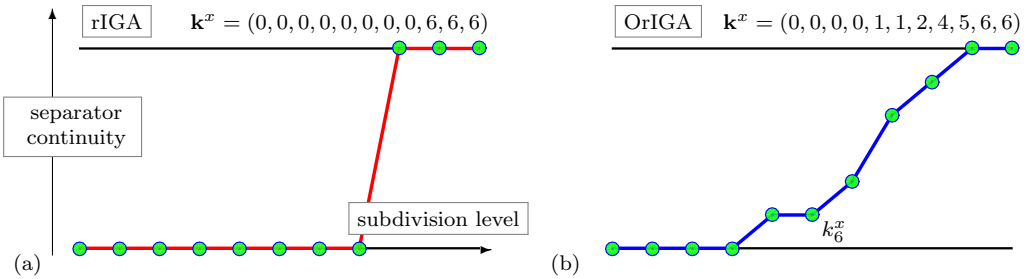


Figure 2: Continuity histograms of the separators for (a) rIGA and (b) OrIGA for $s = 11$ subdivision levels for a septic ($p = 7$) spline space. While rIGA considers separators of only minimum (0) and maximum ($p-1$) continuities, OrIGA explores all admissible continuities $\{0, 1, \dots, p-1\}$ in every subdivision level.

Proof. By contradiction. Let e.g. $k_i^x > k_{i+1}^x$ for some i. We show that there exists $\tilde{\mathbf{k}}$ such that $F(\mathbf{k}) > F(\tilde{\mathbf{k}})$. Define

$$\tilde{\mathbf{k}} = (k_1^x, \dots, k_{i-1}^x, k_{i+1}^x, k_i^x, k_{i+2}^x, \dots, k_s^x, k_1^y, \dots, k_s^y) \tag{13}$$

There are two kinds of summands in (7) that are affected by the switch of k_i^x and k_{i+1}^x. The first type of summand is of the form

$$c_1(c_2 - c_3 k_i^x - 2c_3 k_{i+1}^x), \quad c_1, c_2, c_3 \in \mathbb{N}. \tag{14}$$

Note that these numbers depend on i, s, p_x and p^y. However, all these summands decrease when flipping k_i^x with k_{i+1}^x, that is,

$$k_i^x + 2k_{i+1}^x < k_{i+1}^x + 2k_i^x \tag{15}$$

which is equivalent to $k_i^x > k_{i+1}^x$.

The other types of summands are the two containing terms $(k_i^x + 1)^3$ and $(k_{i+1}^x + 1)^3$, respectively. The sum of these two summands reads as

$$4^{i-1}(k_i^x + 1)^3(2L - c_1)^3 + 4^i(k_{i+1}^x + 1)^3 L^3 \quad c_1, L \in \mathbb{N}. \tag{16}$$

which again decreases with the change of k_i^x and k_{i+1}^x under the assumption that $k_i^x > k_{i+1}^x$. Therefore $F(\mathbf{k}) > F(\tilde{\mathbf{k}})$ which contradicts that $\mathbf{k}$ is the minimizer. □

Lemma 1 introduces a significant reduction of the continuity search space. Observe that while there are p^s possible continuity vectors in one variable, the number of non-decreasing continuity vectors is equal to the number of non-decreasing paths in a rectangular $p \times s$ grid, which is only $\binom{p+s}{s}$, see Fig. 3. For example, for $p = 5$ and $s = 10$, the cardinality of the reduced 1D space is only 3 003 while in the case of the whole search space is 9 765 625. For the 2D case, the reduced space size is $3\,003^2$.

4 Optimally refined Isogeometric Analysis (OrIGA)

Our search for the continuity-aware optimal IGA starts with the rIGA solution. We employ an heuristic approach that uses the following observation from our numerical experiments:

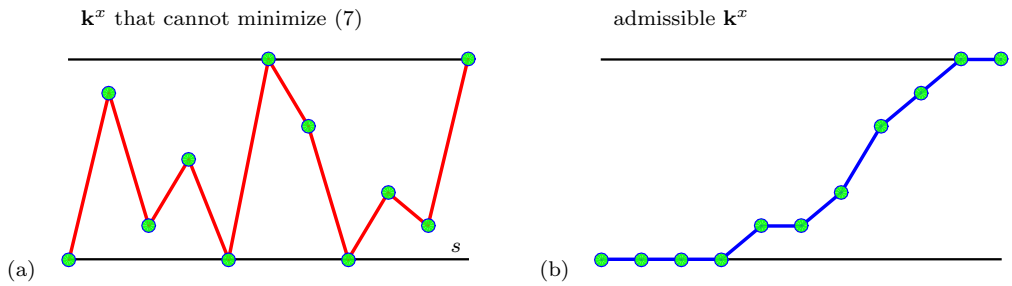


Figure 3: Search space reduction. (a) While the space of all possible continuity vectors grows exponentially in the number of subdivision levels s, see (11), Lemma 1 reduces the search space to only non-decreasing continuity vectors (b).

rIGA and OrIGA solutions are strongly related. Thus, we use rIGA discretization (represented by the continuity vector $\mathbf{k}^{\text{rIGA}}$) to initialize OrIGA and explore exhaustively only a certain neighborhood of $\mathbf{k}^{\text{rIGA}}$. Let i be the number of subdivision levels where rIGA is enriched by C^0-continuous separators (the "jump" of the rIGA continuity vector). We define the *r-neighborhood* of $\mathbf{k}^{\text{rIGA}}$ as the number of subdivision levels that occurred r subdivisions prior i, (and r subdivisions after $i+1$), see Fig. 4. In the r-neighborhood, we consider all continuities that satisfy Lemma 1. Among them, we quickly the find minimizer of (7). If not stated differently, we set $r = 2$ in all our experiments.

Remark 1. *The computation of the rIGA discretization (continuity vector) comes at a negligible cost. Observe that rIGA considers only C^0 and C^{p-1}-continuous separators that are identical in x and y directions, and therefore the computation requires only s evaluations of* (7)*. We denote by $\mathbf{k}^{\text{rIGA}}$ the rIGA solution.*

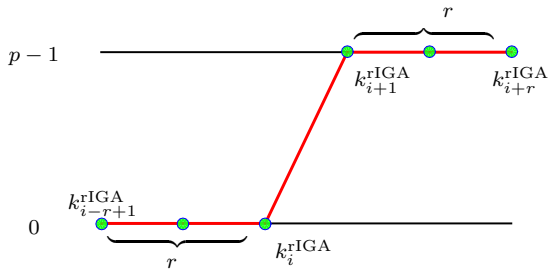


Figure 4: The r-neighborhood of the rIGA solution (red) is shown for $r = 3$. The $2r$ affected continuities of the separators at levels $i-r+1, \ldots, i+r$ are being optimized to minimize (7). The optimal solution must be non-decreasing according to Lemma 1.

Example 1. *An example of rIGA and OrIGA solutions for a septic spline space $p = 7$ over a mesh consisting of $M = 1024^2$ elements is shown in Fig. 5. We searched exhaustively the space S of all feasible continuity vectors, which requires $\binom{7+10}{10}^2 = 19448^2$ evaluations of* (7)*. While the exhaustive search of S required* 507s *on a laptop equipped with a 2.20GHz processor, the computation of rIGA discretization took few milliseconds* ($2.6^{-4}s$) *and the search of its* 2*-neighborhood only* 0.75 *seconds.*

Observe that the OrIGA solution (continuity vector) differs from rIGA only by a few coordinates. This phenomenon applies to various degrees and mesh sizes (see Table 1), where all

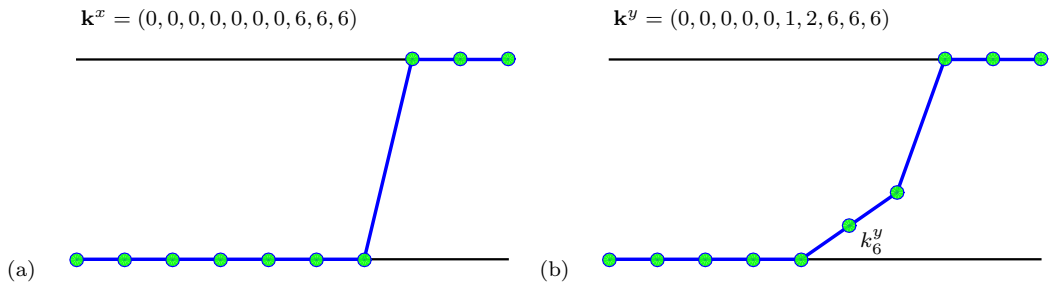


Figure 5: OrIGA: a result of the global exhaustive search in the space separator continuities for a septic spline space ($p = 7$) over a 2D mesh consisting of 1024^2 elements ($s = 10$). The global minimizer $\mathbf{k} = (\mathbf{k}^x, \mathbf{k}^y)$ of (7) is shown. (a) The continuity vector in the x-direction, $\mathbf{k}^x$, is identical to the rIGA solution, while (b) $\mathbf{k}^y$ differs from rIGA by only two coordinates.

Algorithm 1 { OrIGA }

OBJECTIVE: find $\mathbf{k} = (\mathbf{k}^x, \mathbf{k}^y)$ that minimizes (7), i.e., $\arg\min_{\mathbf{k}\in S} F(\mathbf{k})$,
INPUT: number of elements in x and y directions $m = 2^s$;
polynomial degrees p^x and p^y;
search neighborhood r;

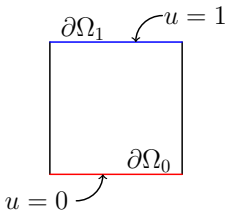
Initialize $\mathbf{k}$ by rIGA solution $\mathbf{k}^{\text{rIGA}}$;
$F_{min} := F(\mathbf{k}^{\text{rIGA}})$;
for $i = 1$ to $\binom{p+2r}{p}$ **do**
  for $j = 1$ to $\binom{p+2r}{p}$ **do**
    build non-decreasing $\mathbf{k}_i^x$ and $\mathbf{k}_j^y$;
    if $F_{min} > F(\mathbf{k}_i^x, \mathbf{k}_j^y)$ **then**
      $\mathbf{k} := (\mathbf{k}_i^x, \mathbf{k}_j^y)$;
      $F_{min} := F(\mathbf{k}_i^x, \mathbf{k}_j^y)$;
    end if
  end for
end for
OUTPUT: OrIGA continuity vector $\mathbf{k}$.

OrIGA and rIGA continuity vectors differ at most at two coordinates. Moreover, this difference appears in the neighborhood of the continuity "jump" of the rIGA continuity vector. Therefore, we use the solution obtained by rIGA to initialize the refined exhaustive search.

Our search for the optimal continuity-aware discretization is summarized in Algorithm 1. Regarding the approximation quality, highly continuous IGA discrete spaces are strictly contained in both the rIGA and OrIGA spaces, so the best approximation error of OrIGA is smaller or equal than that of IGA.

5 Numerical Results

We use Laplace equation as a model problem, see Fig. 6, in our examples. We consider three mesh sizes ($N_e = 512^2$, 1024^2 and 2048^2) and three polynomial degrees ($p = 5$, 7, and 9). The implementation of our method is based on library PetIGA, a high-performance software



$$\begin{cases} \text{Find } u \text{ such that:} & \\ \nabla \cdot (\nabla u) = 0 & \text{in } \Omega \\ u = 1 & \text{on } \partial\Omega_1 \\ u = 0 & \text{on } \partial\Omega_0 \\ \nabla u \cdot n = 0 & \text{on } \partial\Omega_w. \end{cases} \tag{17}$$

Figure 6: Left: Illustration of the 2D model problem. Right: Laplace formulation over parametric domain $\Omega = [0,1]^2$, and $\partial\Omega_w \cup \partial\Omega_0 \cup \partial\Omega_1 = \partial\Omega$, $\partial\Omega_w \cap \partial\Omega_0 = \varnothing$ and $\partial\Omega_w \cap \partial\Omega_1 = \varnothing$.

platform for IGA [6]. We use the multifrontal solver MUMPS [1, 2], and METIS 5 [10] as ordering technique.

The rIGA and OrGA continuity vectors are shown in Table 1. These vectors show that the optimal size of highly continuous macro-elements is almost independent of the mesh size. Namely, it is either 8^2 or 16^2, and then, low-continuity separators should be considered. Table 2 shows FLOPs estimates of FE, IGA, rIGA, and OrIGA for various mesh sizes. We see that both rIGA and OrIGA significantly outperform FE and IGA, OrIGA having expected superior results.

N_e	p	rIGA Continuity	OrIGA Continuity	
512^2	5	[0.0.0.0.0.0.4.4.4.]	$[0.0.0.0.0.0.4.4.4.]\|_x$	$[0.0.0.0.0.2.4.4.4.]\|_y$
	7	[0.0.0.0.0.0.6.6.6.]	$[0.0.0.0.0.0.6.6.6.]\|_x$	$[0.0.0.0.0.2.6.6.6.]\|_y$
	9	[0.0.0.0.0.0.8.8.8.]	$[0.0.0.0.0.0.8.8.8.]\|_x$	$[0.0.0.0.0.1.8.8.8.]\|_y$
1024^2	5	[0.0.0.0.0.0.0.4.4.4.]	$[0.0.0.0.0.0.0.4.4.4.]\|_x$	$[0.0.0.0.0.1.3.4.4.4.]\|_y$
	7	[0.0.0.0.0.0.0.6.6.6.]	$[0.0.0.0.0.0.0.6.6.6.]\|_x$	$[0.0.0.0.0.1.2.6.6.6.]\|_y$
	9	[0.0.0.0.0.0.8.8.8.8.]	$[0.0.0.0.0.0.0.8.8.8.]\|_x$	$[0.0.0.0.0.1.2.8.8.8.]\|_y$
2048^2	5	[0.0.0.0.0.0.0.4.4.4.4.]	$[0.0.0.0.0.0.0.1.4.4.4.]\|_x$	$[0.0.0.0.0.0.2.4.4.4.4.]\|_y$
	7	[0.0.0.0.0.0.0.0.6.6.6.]	$[0.0.0.0.0.0.0.1.6.6.6.]\|_x$	$[0.0.0.0.0.0.1.4.6.6.6.]\|_y$
	9	[0.0.0.0.0.0.0.8.8.8.8.]	$[0.0.0.0.0.0.0.0.8.8.8.]\|_x$	$[0.0.0.0.0.0.1.3.8.8.8.]\|_y$

Table 1: Continuity vectors of rIGA and OrIGA in 2D for various degrees and mesh sizes. OrIGA continuity vectors vary from their corresponding rIGA counterparts, at most, by two values (coordinates) and this variance appears nearby the discontinuity jump of rIGA.

Table 3 shows the actual FLOP counts and computational times using MUMPS. All computational tests were solved sequentially on TACC Lonestar5 system with 2.3 GHz cores and 512TB of memory (URL: http://www.tacc.utexas.edu). Observe that in all cases, except for $p = 5$, $N_e = 2048^2$, OrIGA offers the best discretization.

6 Conclusion and Future Work

We introduce a new discretization method, called OrIGA, that leads to systems of linear equations that can be more efficiently solved via direct solvers. We extend the recently introduced refined isogeometric analysis (rIGA) by considering separators of arbitrary continuities. Our numerical results show that OrIGA, when compared to rIGA, reduces the total computational cost needed by the direct solver by up to 25%, while the computation of the optimal discretization takes only a few seconds. When compared to IGA and FE, we obtain a time boost by factor of up to 60. As a future research direction, we aim to further apply this methodology to non-tensor product meshes.

		$N_e = 512^2$	$N_e = 1024^2$	$N_e = 2048^2$
$p = 5$	FEA	3.46E+11	2.73E+12	2.16E+13
	IGA	3.75E+11	2.87E+12	2.23E+13
	rIGA	1.33E+10	8.97E+10	6.27E+11
	OrIGA	1.26E+10	7.88E+10	5.05E+11
$p = 7$	FEA	1.00E+12	7.69E+12	6.02E+13
	IGA	1.09E+12	8.13E+12	6.23E+13
	rIGA	3.07E+10	1.81E+11	1.18E+12
	OrIGA	2.99E+10	1.67E+11	9.98E+11
$p = 9$	FEA	2.36E+12	1.73E+13	1.32E+14
	IGA	2.32E+12	1.74E+13	1.33E+14
	rIGA	7.03E+10	3.47E+11	1.68E+12
	OrIGA	4.91E+10	2.69E+11	1.61E+12

Table 2: Estimates of the number of FLOPs required by LU factorization. Values obtained with (7), when applied to the tested discretizations with various degrees and mesh sizes.

		$N_e = 512^2$		$N_e = 1024^2$		$N_e = 2048^2$	
		FLOPS	time	FLOPS	time	FLOPS	time
$p = 5$	FEA	$3.479E11$	36.2	$2.937E12$	232.9	$***$	$***$
	IGA	$3.012E11$	22.5	$2.499E12$	162.5	$2.214E13$	1310.9
	rIGA	$1.384E10$	2.8	$1.012E11$	13.7	$5.311E11$	56.5
	OrIGA	$1.362E10$	2.4	$8.287E10$	12.5	$5.586E11$	59.7
$p = 7$	FEA	$9.713E11$	86.2	$***$	$***$	$***$	$***$
	IGA	$8.309E11$	53.9	$6.945E12$	405.9	$5.812E13$	3204.3
	rIGA	$2.960E10$	5.9	$1.914E11$	24.9	$1.323E12$	129.7
	OrIGA	$2.883E10$	5.1	$1.713E11$	21.7	$1.060E12$	104.7
$p = 9$	FEA	$***$	$***$	$***$	$***$	$***$	$***$
	IGA	$1.652E12$	102.1	$1.409E13$	820.1	$1.195E14$	6657.8
	rIGA	$8.163E10$	10.1	$3.806E11$	44.0	$***$	$***$
	OrIGA	$6.990E10$	8.6	$3.244E11$	36.7	$***$	$***$

Table 3: The actual number of FLOPs and computational times required by MUMPS to factorize the 2D problem. The asterisks reflect that the computation was not accomplished due to the lack of memory.

Acknowledgments David Pardo has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 644602, the Projects of the Spanish Ministry of Economy and Competitiveness with reference MTM2013-40824-P, MTM2016-76329-R, and MTM2016-81697-ERC, the BCAM "Severo Ochoa" accreditation of excellence SEV-2013-0323, and the Basque Government through the BERC 2014-2017 program, the Consolidated Research Group Grant IT649-13 on "Mathematical Modeling, Simulation, and Industrial Applications (M2SI)", and the ICERMAR Project KK-2015/0000097.

References

[1] Patrick R. Amestoy, Iain S. Duff, Jean-Yves LÉxcellent, and Jacko Koster. A Fully Asynchronous Multifrontal Solver Using Distributed Dynamic Scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.

[2] Patrick R. Amestoy, Abdou Guermouche, Jean-Yves LÉxcellent, and Stéphane Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing*, 32(2):136 – 156, 2006.

[3] F. Auricchio, L. Beiro da Veiga, A. Buffa, C. Lovadina, A. Reali, and G. Sangalli. A fully locking-free isogeometric approach for plane linear elasticity problems: A stream function formulation. *Computer Methods in Applied Mechanics and Engineering*, 197(1-4):160 – 172, 2007.

[4] N. Collier, D. Pardo, L. Dalcin, M. Paszynski, and V.M. Calo. The cost of continuity: A study of the performance of isogeometric finite elements using direct solvers. *Computer Methods in Applied Mechanics and Engineering*, 213-216(0):353 – 361, 2012.

[5] J. A. Cottrell, T. J. R. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 196(4144):4160 – 4183, 2007.

[6] L. Dalcin, N. Collier, P. Vignal, A. Cortes, and V.M. Calo. PetIGA: A framework for high-performance isogeometric analysis. *arxiv*, (1305.4452), 2015. http://arxiv.org/abs/1305.4452v3.

[7] R Dimitri, L De Lorenzis, MA Scott, P Wriggers, RL Taylor, and G Zavarise. Isogeometric large deformation frictionless contact using t-splines. *Computer methods in applied mechanics and engineering*, 269:394–414, 2014.

[8] D. Garcia, D. Pardo, L. Dalcin, M. Paszynski, N Collier, and V.M. Calo. The value of continuity: Refined isogeometric analysis and fast direct solvers. *Computer Methods in Applied Mechanics and Engineering*, 2016. In press.

[9] H. Gómez, V.M. Calo, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of the Cahn-Hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering*, 197(4950):4333 – 4352, 2008.

[10] Karypis Laboratory. METIS, http://glaros.dtc.umn.edu/gkhome/metis/metis/overview.

[11] Dominik Schillinger, John A Evans, Felix Frischmann, René R Hiemstra, Ming-Chen Hsu, and Thomas JR Hughes. A collocated c0 finite element method: Reduced quadrature perspective, cost comparison with standard finite elements, and explicit structural dynamics. *International Journal for Numerical Methods in Engineering*, 102(3-4):576–631, 2015.

[12] Edward L. Wilson. The static condensation algorithm. *International Journal for Numerical Methods in Engineering*, 8(1):198 – 203, 1974.

[13] J. Yan, A. Korobenko, X. Deng, and Y. Bazilevs. Computational free-surface fluid–structure interaction with application to floating offshore wind turbines. *Computers & Fluids*, 141:155–174, 2016.